PLANT DISEASE DETECTION USING CONVOLUTIONAL NEURAL NETWORKS

A project report submitted for the partial fulfillment of the Bachelor of Technology Degree in Information Technology under Maulana Abul Kalam Azad University of Technology

BY

RISHIKA ROYCHOUDHURY

(ROLL NO: 10400216081 , REGISTRATION NO: 161040110291)

Under the Guidance of:

Prof. Arup Kumar Chattopadhyay

Department of Information Technology

For the Academic Year 2019-2020

Institute of Engineering & Management
Y-12, Salt Lake, Sector-V, Kolkata-700091

Affiliated To:

Maulana Abul Kalam Azad University of Technology

BF-142, Salt Lake,Sector I,Kolkata-700064

CERTIFICATE

TO WHOM IT MAY CONCERN

This is to certify that the project report entitled “Plant Disease Classification using Convolutional Neural Network”, submitted by **Rishika Roychoudhury**, **Roll No: 10400216081**, **Registration Number: 161040110291**, student of **INSTITUTE OF ENGINEERING & MANAGEMENT** in partial fulfillment of requirements for the award of the degree of **Bachelor of Technology in Information Technology**, is a bona fide work carried out under the supervision of **Prof. Arup Kumar Chattopadhyay** during the final year of the academic session of 2016-2020. The content of this report has not been submitted to any other university or institute for the award of any other degree.

Prof . Arup Kr. Chattopadhyay
Project Guide

Dept. Of Information Technology

Institute of Engineering & Management

Prof. Indraneel Mukhopadhyay

H.O.D

Dept. Of Information Technology

Institute of Engineering &Management

Prof.Dr.A.K.Nayak

Principal

Institute of Engineering & Management

Sector-V, Salt Lake Electronics Complex, Kolkata-700091

ACKNOWLEDGEMENT

I would like to take this opportunity to extend our gratitude to the following revered persons without whose immense support, completion of this project wouldn’t have been possible.

I are sincerely grateful to our advisor and mentor **Prof. Arup Kumar Chattopadhyay** of the Department of **Information Technology**, IEM Kolkata, for his constant support , significant insights and for generating in us a profound interest for this subject that kept us motivated during the entire duration of this project.

I would also like to express our sincere gratitude to **Prof. Dr. Satyajit Chakrabarti** (Director**,** IEM)**, Prof. Dr. Amlan Kusum Nayak (**Principal, IEM) and **Prof. Indraneel Mukhopadhyay** HOD of **Information Technology** and other faculties of Institute of Engineering & Management, for their assistance and encouragement.

Last but not the least, I would like to extend our warm regards to our families and peers who have kept supporting us and always had faith in our work.

Rishika Roychoudhury

Reg.No: 161040110291
Dept. of Information Technology

Institute of Engineering & Management, Kolkata

DECLARATION

FOR NON-COMMITMENT OF PLAGIARISM

I , Rishika Roychoudhury,student of Bachelor of Technology in Information Technology, Institute of Engineering & Management have submitted the project report in partial fulfillment of the requirements to obtain the above noted degree. I declare that I have not committed plagiarism in any form or violated copyright while writing the report and have acknowledged the sources and/or the credit of other authors wherever applicable. If subsequently it is found that we have committed plagiarism or violated copyright, then the authority has full right to cancel/reject/revoke our degree.

Name of the Student: RISHIKA ROYCHOUDHURY

Full Signature:________________________________

ABSTRACT

Agriculture has always been an important sector for India as a whole . Plant Production is always a factor for knowing the economic growth of the country. Plants are affected by different types of virus and bacterial attacks which can cause decrease in plant production which in turn will effect the economy of the country. It is observed that usually the farmers try to recognize plant diseases through their naked eyes which is a difficult job and a time taking process. Moreover,detection of diseases aren’t accurate. Image Processing is a medium to solve this problem in fast & effective manner. It help to analyze the images and give the desired results. My project is an optimized approach for detection of plant diseases using convolutional neural networks. All the major project requirements are described in my project report starting from collecting of datasets to getting the desired deep learning frame work. The model will help us to distinguish between diseased leaves and healthy leaves

Keywords: Deep Learning,Image Processing,detection,accurate,Convolutional Neural Network

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

Overview:

Food security is always a major issue in our country which may occur due to different type of plant diseases. In the present scenario,the most challenging situation is precisely diagonosing various diseases. In the Traditional System,human interaction by visual interaction was the only medium to identify it but presently there are many techniques to detect it like polymerase chain reaction,fluorescence in situ hybridization,enzyme-linked immunosorbent assay and many more used to detect plant disease due to bacterial pathogens.

Due to improvement in technology with time,image -based diagonisis has been started. The acquired images contains condensed information which is not easy for the computer to process,there is a requirement of pre-processing certain features.eg:color and shape. To allow the computer to learn suitable features autonomously deep learning is used. There was an initial attempt in 2016 where a trained model classified 14 crops and 26 diseases with an accuracy of 99.35% against optical images. Since then,deep learning based diagnosis is started from different type of crops.

CNNs consist of convolutional layers, which are sets of image filters convoluted to images or feature maps, along with other (e.g., pooling) layers. Convolution is a specialized kind of linear operation. Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers. CNN – as we know its a machine learning algorithm for machines to understand the features of the image with foresight and remember the features to guess whether the name of the new image fed to the machine. In image classification, feature maps are extracted through convolution and other processing layers repetitively and the network eventually outputs a label indicating an estimated class. Given a training dataset, CNN, unlike traditional machine learning techniques that use *hand-crafted* features, optimizes the weights and filter parameters in the hidden layers to generate features suitable to solve the classification problem. In principle, the parameters in the network are optimized by back-propagation and gradient descent approaches to minimize the classification error.

Most efforts utilizing deep learning for disease detection have trained and tested their efforts using the Plant Village dataset, which consists of images with low variability and similar backgrounds. Recently, Barbedo[1] investigated the performance of deep learning models when trained using individual lesions and spots, using image segmentation and augmentation to increase the size of the dataset from a relatively small number of images. They concluded that the performance of the models improved (12% higher on average) in every plant type under investigation with accuracies consistently above 75% in the most complex case with 10 diseases. This research work follows a similar approach: the image datasets are segmented to use maximum information about the disease symptoms by extracting affected parts of the leaves instead of whole leaf images. By quantifying the results of the two models and comparing the self-classification confidence for each disease type, it is shown that we can train using more meaningful data and very good results can be achieved even in real world conditions.

Objective:

Proper pesticides control and remedies can help us to control the attack of pests .The size of images can be reduced by proper implementation of image reduction techniques.During this process,sizes shouldn't be compromised to that extent as it will lead to loss of data due to which accuracy can be affected. My main objective is to expand the project research done by previous autors to give an optimised solution for detection of plant diseases. Image Processing is used as a medium to detect plant diseases. These reduced size images are used as a input dataset in our training model. CNN is used to solve this problem. As a result of this project,I am able to identify various plant diseases.

Chapter 2

BACKGROUND

Literature Review:

In [2],classification and detection methods for plant disease classification has been presented.Pre-processing is done before feature extraction. All RGB images are converted firstl y into white and further into grey colored images . This process is mainly done to mainly extract the vein of the images of leaves. After this step,basic morphological functions are applied on the images.Furthermore,images converted into binary images. After that if binary pixel value is 0 then its is converted into corresponding RGB image value. Finally,pearson correlation and dominating feature set and Naive Bayesian classifier are used to classify plant diseases.

Padol et al. [3],proposed four steps. Initially for training and testing,images are gathered from different parts of the countries. After that,Gaussian filter is used to remove all type of noise and thresholding is used to remove all other components other than all green component. Segmentation is done by K-means clustering algorithms. For extracting features, all RGB images are converted into HSV images.

In [4] ,detection of jute plant disease is done using image processing. After preprocessing the images Hue based segmentation is done with customized thresholding formulae. Further ,RGB images are converted into HSV images which helps to identify different features.this approach helped significantly to identify stem oriented diseases in jute plant.

Tejoindhi et al.[5],mainly proposed a technique which can be used to detect paddy plant disease. In this approach comparison of 100 healthy images and 100 sample of disease1 images and disease2 images is done. This method is neither sufficient to detect the disease nor classifying its training data which is not linearly separable.

In [6] , a method of detection of plant disease in which firstly images are converted from RGB to HSI format,green pixels are removed using masking. Using Ostu’s method,segment the components. Texture features are computed using color color-co-occurrence model. Finally disease is classified using Genetic algorithm.

In [7],a proposal was made on an idea of tomato disease detection using computer vision. Gray scale image converted into binary image depending on threshold value. Image segmentation uses threshold algorithm. Color indices like red,green, blue have their own threshold value. It is not reliable as it is only able to diffrentiate red tomatoes from other colors. Ripe and unripe tomatoes cannot be recognized.K-means algorithm is used to remove this drawback. Non hierarchical clusters are created by K-means Algorithm. This is a unsupervised and non-deterministic method. Then,separation of the infected parts ,the RGB image is converted to YcbCr to improve the features of the image.At last,diffrentiation of ripe and unripe tomatoes is done.

The method for detection of cucumber disease is done in [9]. Pre processing of the images is done which results in feature extraction using Gray level co-occurrence matrix(GLCM) . Finally there are two types of classification :Unsupervised and Supervised classification. Paddy plant is an important plant in continental re-gion. In [9],RGB images are converted into gray scale image using color conversion. Various en-hancement techniques like histogram equalization and contrast adjustment are used for image quality enhancement. Different types of classification features like SVM, ANN, FUZZY classification are used here.

Chapter 3

PRELIMINARIES

Terminologies Used:

1.Dataset:

I found the datasets from spMohanty(https://github.com/spMohanty/PlantVillage-Dataset/trunk/raw/color) and extracted only colored images.

2.Numpy:

NumPy ,as described in [10] ,is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

3.Keras:

Keras,as described in [11], is an open-source neural-network library written in Python. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible.

3.Matplotlib:

Matplotlib[12], is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits

4.Scikit-learn:

Scikit-learn (formerly scikits.learn and also known as sklearn),as described in [13],is a free software machine learning library for the python programming language. It features various classification,regression and clustering algorithms including support vector machine ,random forest,gradient boosting,k-means,and DBSCANand is designed to interoperate with the Python numerical and scientific libraries Numpy and Scipy.

Image Preprocessing and Labelling:

In my project,the first initial step is pre-processing of images for preparing the initial datasets. This step mainly includes removing low-frequency background noise,normalizing the intensity of images,removing reflections,and masking of images. Image Processing is mainly used for enhancing quality of images. This procedure mainly involves cropping of images in order to get the region of interest. Images with less resolution than 500 pixels is not a valid images for datasets. Only the images where the region of interest is more is marked as eligible candidates. In the interest of confirming the accuracy of classes in the dataset, initially grouped by a keywords search, agricultural experts examined leaf images and labeled all the images with appropriate disease acronym.For training and validation it is important to accurately classify images. Only through this ,reliable model can be developed.

Neural Network Training:

In my project,training a deep CNN for creating a picture classification model from a dataset is principally done. CNN may be a kind of feed-forward artificial neural network within which the connectivity pattern between its neurons is inspired by the organization of the animal cortical region. Biological processes are the medium from which CNN is inspired. It has variations of multilayer perceptron which helps in minimization of pre processing. CNN has various forms of application such as image recognition,video recognition,NLP and recommender systems. It is a multi layer system of receptive fields. The input image is processed using these small neuron collections. Furthermore,output of those neuron collections are tiled to create the input data overlap to induce a high resolution of the initial image. With the assistance of tiling CNNs can tolerate the interpretation of the input image. Combination of output of neuron clusters is completed by local orglobal pooling layers. To reduce the quantity of free parameters and improve generalization a convolution operation on small regions of input is introduced. One of the key advantage of convolutional neural networks is that the use of shared weight , which suggests that the identical filter (weights bank) is employed for every pixel within the layer,this leads to reduction of both reduces memory footprint and improve performance. Rectified Linear Units (Re LU) are used as there's a substitute for saturating nonlinearities,Rectified Linear Units . This activation function adaptively learns the parameters of rectifiers and improves accuracy at negligible extra computational cost,within the context of artificial neural networks, the rectifier is an activation function defined as:

$f(x)=\max(0,x)$, where x is the input to a neuron. This is also known as a ramp function and is analogous to half-wave rectification in electrical engineering. It has been utilized in convolutional networks more effectively than the widely used logistic sigmoid (which is inspired by probability theory; see logistic regression) and its more practical counterpart, the hyperbolic tangent. The rectifier is, as of 2015, the foremost popular activation function for deep neural networks. Deep CNN with ReLUs trains several times faster. This method is applied to the output of each convolutional and fully connected layer. Despite the output, the input normalization isn't required; it's applied after ReLU nonlinearity after the primary and second convolutional layer because it reduces top-1 and top-5 error rates. In CNN, neurons within a hidden layer are segmented into “feature maps”. The neurons within a feature map share the identical weight and bias. The neurons within the feature map hunt for the identical feature. These neurons are unique since they're connected to different neurons within the lower layer. So for the primary hidden layer, neurons within a feature map are connected to different regions of the input image. The hidden layer is segmented into feature maps where each neuron in an exceedingly feature map looks for the identical feature but at different positions of the input image. Basically, the feature map is that the results of applying convolution across a picture. The convolutional layer is that the core building block of a CNN. The layer's parameters include a collection of learnable filters (or kernels), which have alittle receptive field, but extend through the complete depth of the input volume. During the pass, each filter is convolved across the width and height of the input volume, computing the real between the entries of the filter and also the input and producing a 2-dimensional activation map of that filter. As a result, the network learns filters that activate when it detects some specific kind of feature at some spatial position within the input. Stacking the activation maps for all filters along the depth dimension forms the complete output volume of the convolution layer. Every entry within the output volume can thus even be interpreted as an output of a neuron that appears at alittle region within the input and shares parameters with neurons within the same activation map. When coping with high-dimensional inputs like images, it's impractical to attach neurons to all or any neurons within the previous volume because such spec doesn't take the spatial structure of the info into consideration. Convolutional networks exploit spatially local correlation by enforcing an area connectivity pattern between neurons of adjacent layers: each neuron is connected to only alittle region of the input volume. The extent of this connectivity may be a hyper parameter called the receptive field of the neuron. The connections are local in space (along width and height), but always extend along the whole depth of the input volume. Such architecture ensures that the learnt filters produce the strongest response to a spatially local input pattern. Three hyper parameters control the dimensions of the output volume of the convolutional layer: the depth, stride and zero-padding.

1. Depth of the output volume controls the amount of neurons within the layer that hook up with the identical region of the input volume. All of those neurons will learn to activate for various features within the input. or instance, if the primary Convolutional Layer takes the raw image as input, then different neurons along the depth dimension may activate within the presence of varied oriented edges, or blobs of color.

2. Stride controls how depth columns round the spatial dimensions (width and height) are allocated. When the stride is 1, a replacement depth column of neurons is allocated to spatial positions only one spatial unit apart. This ends up in heavily overlapping receptive fields between the columns, and also to large output volumes. Conversely, if higher strides are used then the receptive fields will overlap less and also the resulting output volume will have smaller dimensions spatially.

3. Stride controls how depth columns round the spatial dimensions (width and height) are allocated. When the stride is 1, a replacement depth column of neurons is allocated to spatial positions only one spatial unit apart. This ends up in heavily overlapping receptive fields between the columns, and also to large output volumes. Conversely, if higher strides are used then the receptive fields will overlap less and also the resulting output volume will have smaller dimensions spatially.

Parameter sharing scheme is employed in convolutional layers to regulate the amount of free parameters. It relies on one reasonable assumption: That if one patch feature is helpful to compute at some spatial position, then it should even be useful to compute at a special position. In other words, denoting one 2-dimensional slice of depth as a depth slice, we constrain the neurons in each depth slice to use the identical weights and bias. Since all neurons in an exceedingly single depth slice are sharing the identical parameterization, then the passing game in each depth slice of the CONV layer will be computed as a convolution of the neuron's weights with the input volume (hence the name: convolutional layer). Therefore, it's common to visit the sets of weights as a filter (or a kernel), which is convolved with the input. The results of this convolution is an activation map, and also the set of activation maps for every different filter are stacked together along the depth dimension to supply the output volume. Parameter Sharing contributes to the interpretation invariance of the CNN architecture. it's important to note that sometimes the parameter sharing assumption might not add up. this can be especially the case when the input images to a CNN have some specific centred structure, within which we expect completely different features to be learned on different spatial locations. One practical example is when the input is faces that are centred within the image: we would expect different eye-specific or hair-specific features to be learned in numerous parts of the image. therein case it's common to relax the parameter sharing scheme, and instead simply call the layer a locally connected layer. Another important layer of CNNs is that the pooling layer, which may be a kind of nonlinear down sampling.

Pooling operation gives the shape of translation invariance. It operates independently on every depth slice of the input and resizes it spatially. Overlapping pooling is beneficially applied to minimize over fitting. Also in favour of reducing over fitting, a dropout layer is employed within the first two fully connected layers. But the shortcoming of dropout is that it increases training time 2-3 times comparing to a typical neural network of the precise architecture. Bayesian optimization experiments also proved that ReLUs and dropout have synergy effects, which implies that it's advantageous once they are used together. The advance of CNNs refers to their ability to find out rich mid-level image representations as against hand-designed low-level features employed in other image classification methods.

Chapter 3

PROPOSED STRATEGY

Proposed scheme:

Step1: Firstly ,loading the datset (image of diseased plants) is performed. Conversion of each image labels to binary levels usking sk-learn Label Binarizer. Furthermore,pre-processing of input data where scaling the data points from[0,255](Maximum and minimum RGB values of the image) to the range[0,1]. After this steps,splitting of data into 80% of the images in training and 20% for validation is done.

Step2: Image generator object is created which performs flips,crops,sheer and random rotations in the given datasets. After this,a convolutional layer is created . It has 32 filters with 3*3 kernels and then Rectified Linear Unit activation. Then we apply batch normalization,0.25 dropout and max pooling.

Dropout is a regularization technique for reducing overfitting in neural networks by preventing complex co-adaptations on training data. It is a very efficient way of performing model averaging with neural networks.[14]

Step3: Two set of (CONV => RELU) * 2 => POOL blocks is created. Then only one set of FC (Fully Connected Layer)=> RELU layers.

Step4: Then further,I created two sets of (CONV => RELU) * 2 => POOL blocks. Training our network is initiated where we call model.fit_generator , supplying our data augmentation object, training/testing data, and the number of epochs we wish to train for. I used an epochs value of 25 for this project.

Step5: Then plotting of graph using matplotlib for Training and Validation accuracy and Training and Validation loss graph.

Step6: Then calculation of accuracy of the model using the validation data (x_test and y_test) created earlier. The accuracy received an accuracy score of 96.68%.

Chapter 4

EXPERIMENTAL SETUP AND PRACTICAL IMPLEMENTATION

Experiment Results:

In Figure 1,Various inputs from Kaggle Datasets(Plant Village). These Dataset includes various images of healthy and unhealthy leaves of different plants.

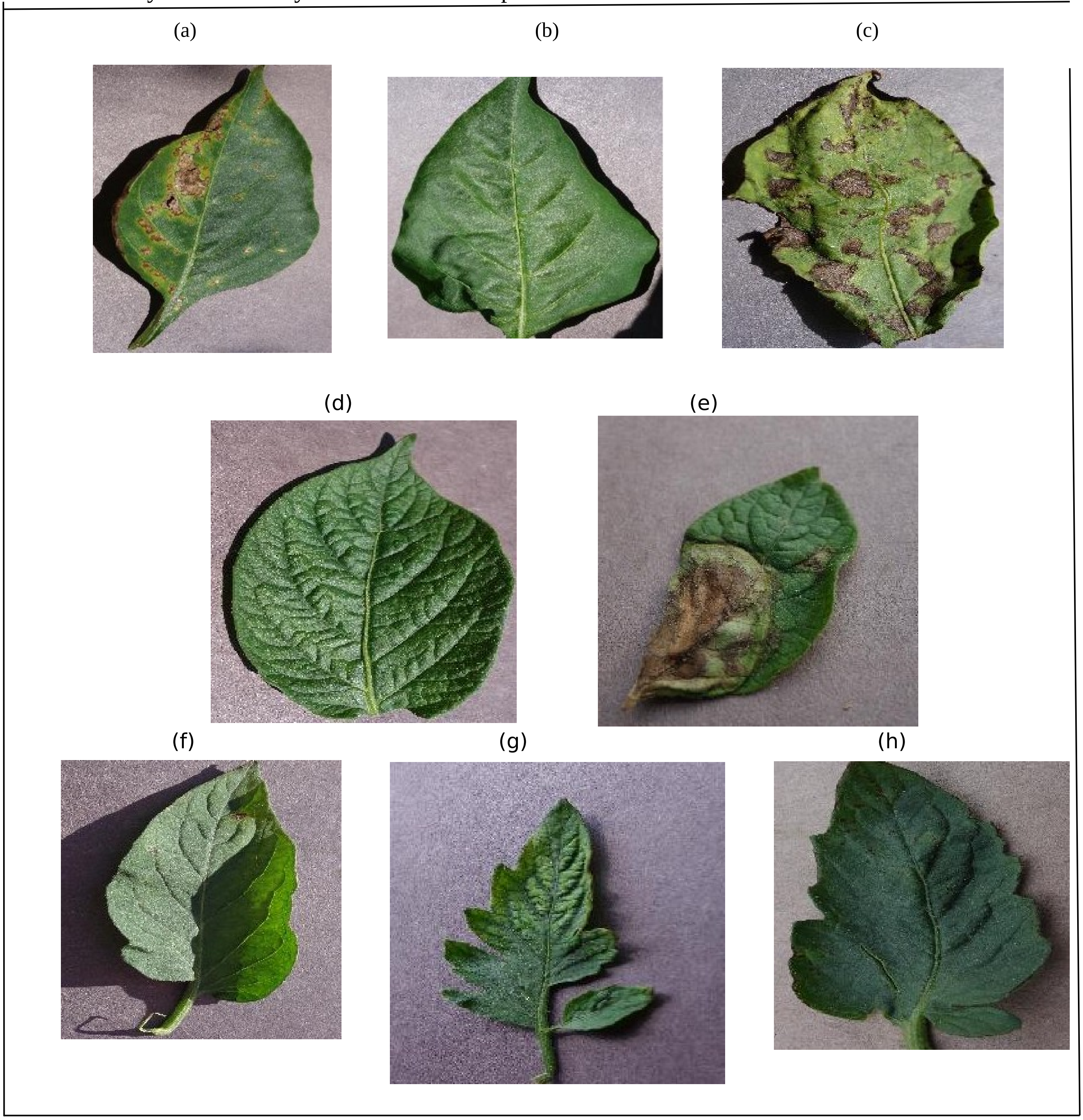

Figure 1:Images of Sample Dataset(Kaggle:Plant Village)

In Figure2 ,with the help of Sckit-learn I have converted each and every image label to binary levels

```
['Pepper__bell___Bacterial_spot' 'Pepper__bell___healthy'
 'Potato___Early_blight' 'Potato___Late_blight' 'Potato___healthy'
 'Tomato_Bacterial_spot' 'Tomato_Early_blight' 'Tomato_Late_blight'
 'Tomato_Leaf_Mold' 'Tomato_Septoria_leaf_spot'
 'Tomato_Spider_mites_Two_spotted_spider_mite' 'Tomato__Target_Spot'
 'Tomato__Tomato_YellowLeaf__Curl_Virus' 'Tomato__Tomato_mosaic_virus'
 'Tomato_healthy']
```

Figure2:Converting the image labels to binary using Scikit-learn's Label Binarizer (classes of Label Binarizer)

In Figure 3,it is showing the details of training and validation samples after splitting the datasets.

```
Training samples in Tomato__Target_Spot is 1123
Validation samples in Tomato__Target_Spot is 281

Training samples in Pepper__bell___healthy is 1182
Validation samples in Pepper__bell___healthy is 296

Training samples in Potato___healthy is 122
Validation samples in Potato___healthy is 30

Training samples in Tomato__Tomato_mosaic_virus is 298
Validation samples in Tomato__Tomato_mosaic_virus is 75

Training samples in Potato___Early_blight is 800
Validation samples in Potato___Early_blight is 200

Training samples in Tomato__Tomato_YellowLeaf__Curl_Virus is 2567
Validation samples in Tomato__Tomato_YellowLeaf__Curl_Virus is 642
```

Figure 3:Showing the details of unhealthy leaves(Training and Validation samples details)

Accuracy of the trained Model: Accuracy plays an important role to determine whether the trained model will be suitable or whether it provide us with desired outputs or not. After splitting the datasets training 80% and testing 20% of datasets accuracy obtained is shown in Figure4.

Calculating the Accuracy of the trained model by the validated datasets:

```
[INFO] Calculating model accuracy
591/591 [==============================] - 1s 2ms/step
Test Accuracy: 96.68358961741129
```

Figure 4:Accuracy obtained through my proposed model

Figure 5 shows the training and validation accuracy graph obtained by my proposed scheme

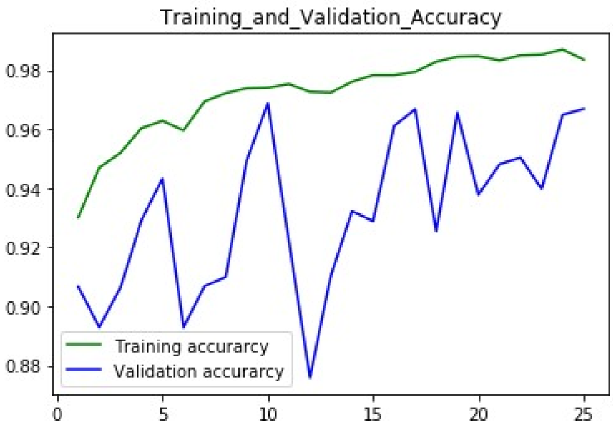

Figure 5:Training and Validation Accuracy graph

In Figure 5,According to the observations,validation accuracy occurs to be less than training accuracy. This can be improved by reshuffling the validation set . The model should be trained more for validation samples as the datasets in that case are still rising and falling for some epochs.

In Figure 6, Representation of training and validation loss graph is shown.

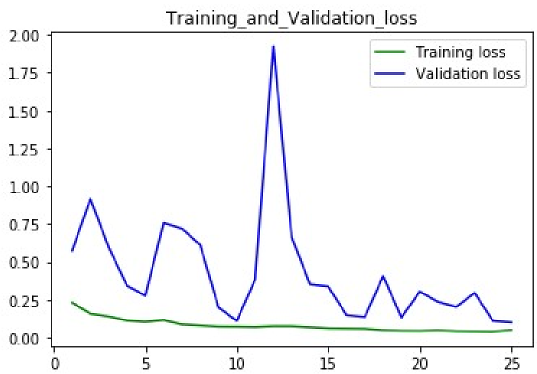

Figure 6:Training and Validation Loss Graph

Training and Validation Loss graph is mainly used to show whether the model have comparable performance or not. When the parallel plots depart consistently after some time it is a flag step to stop our training at an earlier epoch.

In Figure 6,it is observed that there is higher validation loss than training loss,it represents that my model is overfitting. It learns superstition which means that patterns in my datasets of training data but in reality its not the case and hence it is not true for validation data.

Overfitting mainly occurs when the function is too closely fit to limited set of data points. It takes the form of making an overly complex model to explain idiosyncrasies in the data under study.

Chapter 5

CONCLUSION

Conclusion:

This proposed system will help farmers and will be a great contribution to agricultural sector. It can detect plant diseases and correspondingly the remedy for the disease. I have used Python as the programming language to compute the problem statement. The proposed system has accuracy of 96.68. I have used Kaggle kernels due to increased speed for processing the data. It helps in automatic detection of diseases as well as classifying these plant diseases. This method is tested in for Jute,Grape,Paddy,Okra,Tomatoes.Optimum results are obtained by very minimum computational efforts. It also shows the efficiency of algorithm in recognition and classification of the leaf diseases. This present method isn't completely exploited in the field of plant disease recognition.

Future Improvements:

Sustainable plant disease management system can be created which can impact economics,ecology and sociology. This model can help the farmers not only to enhance their agricultural productivity but also improve food quality which can protect our natural environment. To achieve the goal their can be following improvements:

i) Plant disease epidemic can be analyzed properly

ii) Real time based integrated system can be developed of major plant diseases.

Iii) Change in pattern of environmental conditions should be taken into count for getting accurate data.

Chapter 6

REFERENCES

References:

[1] Barbedo, Jayme Garcia Arnal. "Plant disease identification from individual lesions and spots using deep learning." *Biosystems Engineering* 180 (2019): 96-107

[2] Mondal, D., Chakraborty, A., Kole, D.K. and Majumder, D.D., 2015, October. Detection and classification technique of yellow vein mosaic virus disease in okra leaf images using leaf vein extraction and Naive Bayesian classifier. In *2015 International Conference on Soft Computing Techniques and Implementations (ICSCTI)* (pp. 166-171). IEEE.

[3] Padol, P.B. and Yadav, A.A., 2016, June. SVM classifier based grape leaf disease detection. In *2016 Conference on advances in signal processing (CASP)* (pp. 175-179). IEEE.

[4] Reza, Z.N., Nuzhat, F., Mahsa, N.A. and Ali, M.H., 2016, September. Detecting jute plant disease using image processing and machine learning. In *2016 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)* (pp. 1-6). IEEE.

[5] Tejoindhi, M.R. and Nanjesh, B.R., JagadeeshGujanuru Math. In *AshwinGeetD'sa" Plant Disease Analysis Using Histogram Matching Based On Bhattacharya's Distance Calculation" International Conference on Electrical, Electroniocs and Optimization Techniques (ICEEOT)-2016*.

[6] Arya, M.S., Anjali, K. and Unni, D., 2018, January. Detection of unhealthy plant leaves using image processing and genetic algorithm with Arduino. In *2018 International Conference on Power, Signals, Control and Computation (EPSCICON)* (pp. 1-5). IEEE.

[7] Mehra, T., Kumar, V. and Gupta, P., 2016, December. Maturity and disease detection in tomato using computer vision. In *2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)* (pp. 399-403). IEEE.

[8] Pawar, P., Turkar, V. and Patil, P., 2016, August. Cucumber disease detection using artificial neural network. In *2016 International Conference on Inventive Computation Technologies (ICICT)* (Vol. 3, pp. 1-5). IEEE.

[9] Narmadha, R.P. and Arulvadivu, G., 2017, January. Detection and measurement of paddy leaf disease symptoms using image processing. In *2017 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1-4). IEEE.

[10] https://en.wikipedia.org/wiki/Numpy

[11] https://en.wikipedia.org/wiki/keras

[12] https://en.wikipedia.org/wiki/matplotlib

[13] https://en.wikipedia.org/wiki/Scikit-learn

[14] https://en.wikipedia.org/wiki/Dropout_(neural_networks)